UPPSALA
UNIVERSITET

Energy-Based Probabilistic Regression in Computer Vision

Fredrik K. Gustafsson
Uppsala University

Half-time seminar
February 3, 2022

[Paper I] Energy-Based Models for Deep Probabilistic Regression
Fredrik K. Gustafsson, Martin Danelljan, Goutam Bhat, Thomas B. Schön
The European Conference on Computer Vision (ECCV), 2020

[Paper II] How to Train Your Energy-Based Model for Regression
Fredrik K. Gustafsson, Martin Danelljan, Radu Timofte, Thomas B. Schön
The British Machine Vision Conference (BMVC), 2020

[Paper III] Accurate 3D Object Detection using Energy-Based Models
Fredrik K. Gustafsson, Martin Danelljan, Thomas B. Schön
The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2021

[Paper IV] Learning Proposals for Practical Energy-Based Regression
Fredrik K. Gustafsson, Martin Danelljan, Thomas B. Schön
The International Conference on Artificial Intelligence and Statistics (AISTATS), 2022

Outline

Energy-based models.

Energy-based models for regression.

How to train energy-based models for regression.

- Noise contrastive estimation (NCE).

Energy-based regression for 3D object detection.

Practical limitations of energy-based regression.

Learning proposals for more practical energy-based regression.

Energy-Based Models

Energy-based models have a rich history within machine learning.

An **energy-based model (EBM)** specifies a probability distribution $p(x;\theta)$ over $x \in \mathcal{X}$ directly via a parameterized scalar function $f_\theta : \mathcal{X} \to \mathbb{R}$:

$$p(x;\theta) = \frac{e^{f_\theta(x)}}{Z(\theta)}, \quad Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x}$$

Energy-Based Models

UPPSALA UNIVERSITET

Energy-based models have a rich history within machine learning.

An **energy-based model (EBM)** specifies a probability distribution $p(x;\theta)$ over $x \in \mathcal{X}$ directly via a parameterized scalar function $f_\theta : \mathcal{X} \to \mathbb{R}$:

$$p(x;\theta) = \frac{e^{f_\theta(x)}}{Z(\theta)}, \quad Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x}$$

By defining $f_\theta(x)$ using a deep neural network (DNN), the EBM $p(x;\theta)$ becomes expressive enough to learn practically any distribution from observed data.

Energy-Based Models

Energy-based models have a rich history within machine learning.

An **energy-based model (EBM)** specifies a probability distribution $p(x;\theta)$ over $x \in \mathcal{X}$ directly via a parameterized scalar function $f_\theta : \mathcal{X} \to \mathbb{R}$:

$$p(x;\theta) = \frac{e^{f_\theta(x)}}{Z(\theta)}, \quad Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x}$$

By defining $f_\theta(x)$ using a deep neural network (DNN), the EBM $p(x;\theta)$ becomes expressive enough to learn practically any distribution from observed data.

EBMs have therefore become increasingly popular within computer vision in recent years, commonly being applied for various generative image modeling tasks.

An EBM specifies a probability distribution $p(x;\theta)$ directly via a parameterized scalar function $f_\theta(x)$,

$$p(x;\theta) = \frac{e^{f_\theta(x)}}{Z(\theta)}, \quad Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x},$$

where $f_\theta(x)$ commonly is defined using a DNN.

An EBM specifies a probability distribution $p(x;\theta)$ directly via a parameterized scalar function $f_\theta(x)$,

$$p(x;\theta) = \frac{e^{f_\theta(x)}}{Z(\theta)}, \quad Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x},$$

where $f_\theta(x)$ commonly is defined using a DNN.

The EBM $p(x;\theta) = e^{f_\theta(x)} / \int e^{f_\theta(\tilde{x})} d\tilde{x}$ is thus a highly expressive model that puts minimal restricting assumptions on the true distribution $p(x)$.

An EBM specifies a probability distribution $p(x;\theta)$ directly via a parameterized scalar function $f_\theta(x)$,

$$p(x;\theta) = \frac{e^{f_\theta(x)}}{Z(\theta)}, \quad Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x},$$

where $f_\theta(x)$ commonly is defined using a DNN.

The EBM $p(x;\theta) = e^{f_\theta(x)} / \int e^{f_\theta(\tilde{x})} d\tilde{x}$ is thus a highly expressive model that puts minimal restricting assumptions on the true distribution $p(x)$.

Drawback: the normalizing partition function $Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x}$ is intractable, which complicates evaluating or sampling from the EBM $p(x;\theta)$.

An EBM specifies a probability distribution $p(x;\theta)$ directly via a parameterized scalar function $f_\theta(x)$,

$$p(x;\theta) = \frac{e^{f_\theta(x)}}{Z(\theta)}, \quad Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x},$$

where $f_\theta(x)$ commonly is defined using a DNN.

The EBM $p(x;\theta) = e^{f_\theta(x)} / \int e^{f_\theta(\tilde{x})} d\tilde{x}$ is thus a highly expressive model that puts minimal restricting assumptions on the true distribution $p(x)$.

Drawback: the normalizing partition function $Z(\theta) = \int e^{f_\theta(\tilde{x})} d\tilde{x}$ is intractable, which complicates evaluating or sampling from the EBM $p(x;\theta)$.

Compare with normalizing flow models which are specifically designed to be easy to both evaluate and sample. EBMs instead prioritize maximum model expressivity.

While EBMs recently had become increasingly popular within computer vision, they were basically only being employed for generative image modeling.

In **[Paper I] Energy-Based Models for Deep Probabilistic Regression**, we instead explored the application of EBMs to various regression problems.

While EBMs recently had become increasingly popular within computer vision, they were basically only being employed for generative image modeling.

In **[Paper I] Energy-Based Models for Deep Probabilistic Regression**, we instead explored the application of EBMs to various regression problems.

Regression: learn to predict a continuous target $y^\star \in \mathcal{Y} = \mathbb{R}^K$ from a corresponding input $x^\star \in \mathcal{X}$, given a training set $\mathcal{D}$ of i.i.d. input-target pairs, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, $(x_i, y_i) \sim p(x, y)$.

UPPSALA UNIVERSITET

Regression: learn to predict a continuous target $y^\star \in \mathcal{Y} = \mathbb{R}^K$ from a corresponding input $x^\star \in \mathcal{X}$, given a training set $\mathcal{D}$ of i.i.d. input-target pairs, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, $(x_i, y_i) \sim p(x, y)$.

Regression: learn to predict a continuous target $y^\star \in \mathcal{Y} = \mathbb{R}^K$ from a corresponding input $x^\star \in \mathcal{X}$, given a training set $\mathcal{D}$ of i.i.d. input-target pairs, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, $(x_i, y_i) \sim p(x, y)$.

We employ a probabilistic regression approach, using a *conditional* EBM to model the predictive distribution $p(y|x)$ of the regression target y given the input x:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

UPPSALA UNIVERSITET

Regression: learn to predict a continuous target $y^\star \in \mathcal{Y} = \mathbb{R}^K$ from a corresponding input $x^\star \in \mathcal{X}$, given a training set $\mathcal{D}$ of i.i.d. input-target pairs, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, $(x_i, y_i) \sim p(x, y)$.

We employ a probabilistic regression approach, using a *conditional* EBM to model the predictive distribution $p(y|x)$ of the regression target y given the input x:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

Here, $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a DNN that maps any input-target pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ directly to a scalar $f_\theta(x, y) \in \mathbb{R}$, and $Z(x, \theta)$ is the input-dependent partition function.

Energy-Based Probabilistic Regression

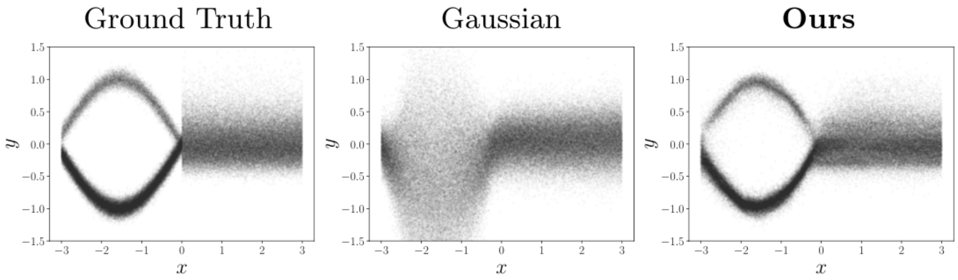
Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x;\theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x;\theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

The EBM $p(y|x;\theta)$ can learn complex distributions $p(y|x)$ directly from data:



Energy-Based Probabilistic Regression

UPPSALA UNIVERSITET

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x;\theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

UPPSALA UNIVERSITET

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x; \theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

We have applied the approach to various regression problems within computer vision:

UPPSALA UNIVERSITET

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x;\theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

We have applied the approach to various regression problems within computer vision:

Paper I:

- Age estimation, $\mathcal{Y} = \mathbb{R}$.
- Head-pose estimation, $\mathcal{Y} = \mathbb{R}^3$.
- 2D bounding box regression (object detection, visual tracking), $\mathcal{Y} = \mathbb{R}^4$.

Energy-Based Probabilistic Regression

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x;\theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

We have applied the approach to various regression problems within computer vision:

Paper I:

- Age estimation, $\mathcal{Y} = \mathbb{R}$.
- Head-pose estimation, $\mathcal{Y} = \mathbb{R}^3$.
- 2D bounding box regression (object detection, visual tracking), $\mathcal{Y} = \mathbb{R}^4$.

Paper II:

- 2D bounding box regression (object detection, visual tracking), $\mathcal{Y} = \mathbb{R}^4$.

UPPSALA UNIVERSITET

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x;\theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

We have applied the approach to various regression problems within computer vision:

Paper III:

- 3D bounding box regression (3D object detection in LiDAR point clouds), $\mathcal{Y} = \mathbb{R}^7$.

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x; \theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

We have applied the approach to various regression problems within computer vision:

Paper III:

- 3D bounding box regression (3D object detection in LiDAR point clouds), $\mathcal{Y} = \mathbb{R}^7$.

Paper IV:

- Steering angle prediction, $\mathcal{Y} = \mathbb{R}$.
- Cell-count prediction, $\mathcal{Y} = \mathbb{R}$.
- Age estimation, $\mathcal{Y} = \mathbb{R}$.
- Head-pose estimation, $\mathcal{Y} = \mathbb{R}^3$.

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x;\theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x;\theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

In **Paper I, II & III**, we predict the most likely target under the model given an input $x^\star$ at test-time, i.e. $y^\star = \text{argmax}_y\, p(y|x^\star;\theta) = \text{argmax}_y\, f_\theta(x^\star, y)$.

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x; \theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

In **Paper I, II & III**, we predict the most likely target under the model given an input $x^\star$ at test-time, i.e. $y^\star = \text{argmax}_y \, p(y|x^\star;\theta) = \text{argmax}_y \, f_\theta(x^\star, y)$.

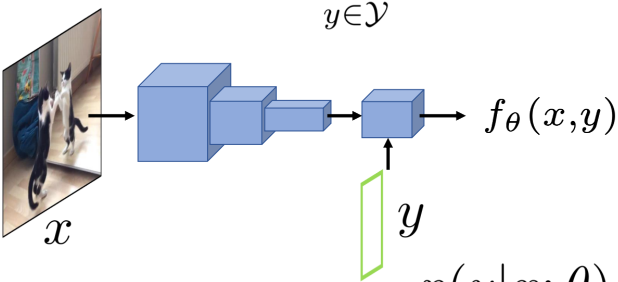
In practice, $y^\star = \text{argmax}_y \, f_\theta(x^\star, y)$ is approximated by refining an initial estimate $\hat{y}$ via T steps of gradient ascent,

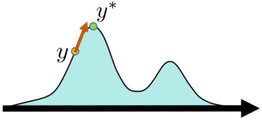
$$y \leftarrow y + \lambda \nabla_y f_\theta(x^\star, y),$$

thus finding a local maximum of $f_\theta(x^\star, y)$. Evaluation of the partition function $Z(x^\star, \theta)$ is therefore *not* required.

Energy-Based Probabilistic Regression - Prediction

$$y^* = \underset{y \in \mathcal{Y}}{\arg\max}\, p(y|x;\theta) = \underset{y \in \mathcal{Y}}{\arg\max}\, f_\theta(x,y)$$





$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{\int e^{f_\theta(x,\tilde{y})}\,\mathrm{d}\tilde{y}}$$

UPPSALA UNIVERSITET

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x; \theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x;\theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

The DNN $f_\theta(x, y)$ can be trained using various methods for fitting a distribution $p(y|x;\theta)$ to observed data $\{(x_i, y_i)\}_{i=1}^N$.

Energy-Based Probabilistic Regression - Training

UPPSALA UNIVERSITET

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x;\theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

The DNN $f_\theta(x, y)$ can be trained using various methods for fitting a distribution $p(y|x;\theta)$ to observed data $\{(x_i, y_i)\}_{i=1}^N$.

In general, the most straightforward such method is probably to minimize the negative log-likelihood $\mathcal{L}(\theta) = -\sum_{i=1}^N \log p(y_i|x_i;\theta)$, which for the EBM $p(y|x;\theta)$ is given by,

$$\mathcal{L}(\theta) = \sum_{i=1}^N \log\Big(\int e^{f_\theta(x_i,y)} dy\Big) - f_\theta(x_i, y_i).$$

Energy-Based Probabilistic Regression - Training

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log p(y_i|x_i;\theta) = \sum_{i=1}^{N} \log\left(\int e^{f_\theta(x_i,y)} dy\right) - f_\theta(x_i, y_i).$$

Energy-Based Probabilistic Regression - Training

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log p(y_i|x_i;\theta) = \sum_{i=1}^{N} \log\left(\int e^{f_\theta(x_i,y)} dy\right) - f_\theta(x_i, y_i).$$

The integral $\int e^{f_\theta(x_i,y)} dy$ is however intractable, preventing exact evaluation of $\mathcal{L}(\theta)$.

UPPSALA UNIVERSITET

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log p(y_i|x_i;\theta) = \sum_{i=1}^{N} \log\left(\int e^{f_\theta(x_i,y)} dy\right) - f_\theta(x_i, y_i).$$

The integral $\int e^{f_\theta(x_i,y)} dy$ is however intractable, preventing exact evaluation of $\mathcal{L}(\theta)$.

In **[Paper I] Energy-Based Models for Deep Probabilistic Regression**, we simply approximated this intractable integral using importance sampling.

Energy-Based Probabilistic Regression - Training

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log p(y_i|x_i;\theta) = \sum_{i=1}^{N} \log\left(\int e^{f_\theta(x_i,y)} dy\right) - f_\theta(x_i, y_i).$$

Importance sampling:

$$\begin{aligned}
-\log p(y_i|x_i;\theta) &= \log\left(\int e^{f_\theta(x_i,y)} dy\right) - f_\theta(x_i, y_i) \\
&= \log\left(\int \frac{e^{f_\theta(x_i,y)}}{q(y)} q(y) dy\right) - f_\theta(x_i, y_i) \\
&\approx \log\left(\frac{1}{M}\sum_{m=1}^{M} \frac{e^{f_\theta(x_i,y^{(m)})}}{q(y^{(m)})}\right) - f_\theta(x_i, y_i), \quad y^{(m)} \sim q(y).
\end{aligned}$$

UPPSALA UNIVERSITET

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x;\theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

Energy-Based Probabilistic Regression - Training

UPPSALA UNIVERSITET

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x; \theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

Various alternative techniques could however also be employed to train the DNN $f_\theta(x, y)$, including noise contrastive estimation (NCE), score matching and MCMC.

UPPSALA UNIVERSITET

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x; \theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

Various alternative techniques could however also be employed to train the DNN $f_\theta(x, y)$, including noise contrastive estimation (NCE), score matching and MCMC.

In **[Paper II] How to Train Your Energy-Based Model for Regression**, we thus studied in detail how EBMs should be trained specifically for regression problems.

UPPSALA UNIVERSITET

Energy-Based Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar $f_\theta(x, y)$, then model $p(y|x)$ with the conditional EBM $p(y|x;\theta)$:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

Various alternative techniques could however also be employed to train the DNN $f_\theta(x, y)$, including noise contrastive estimation (NCE), score matching and MCMC.

In **[Paper II] How to Train Your Energy-Based Model for Regression**, we thus studied in detail how EBMs should be trained specifically for regression problems.

We compared six methods on the task of 2D bounding box regression, and concluded that a simple extension of NCE should be considered the go-to training method.

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

Noise contrastive estimation (NCE) entails learning to discriminate between observed data examples and samples drawn from a noise distribution.

UPPSALA UNIVERSITET

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

Noise contrastive estimation (NCE) entails learning to discriminate between observed data examples and samples drawn from a noise distribution.

Specifically, the DNN $f_\theta(x,y)$ is trained by minimizing $J_{\text{NCE}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} J_{\text{NCE}}^{(i)}(\theta)$,

$$J_{\text{NCE}}^{(i)}(\theta) = \log \frac{\exp\{f_\theta(x_i, y_i^{(0)}) - \log q(y_i^{(0)})\}}{\sum_{m=0}^{M} \exp\{f_\theta(x_i, y_i^{(m)}) - \log q(y_i^{(m)})\}},$$

where $y_i^{(0)} \triangleq y_i$, and $\{y_i^{(m)}\}_{m=1}^{M}$ are M samples drawn from a noise distribution $q(y)$.

Energy-Based Probabilistic Regression - Training using NCE

$$J_{\mathrm{NCE}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} J_{\mathrm{NCE}}^{(i)}(\theta), \quad J_{\mathrm{NCE}}^{(i)}(\theta) = \log \frac{\exp\{f_\theta(x_i, y_i^{(0)}) - \log q(y_i^{(0)})\}}{\sum_{m=0}^{M} \exp\{f_\theta(x_i, y_i^{(m)}) - \log q(y_i^{(m)})\}},$$

$$y_i^{(0)} \triangleq y_i, \quad \{y_i^{(m)}\}_{m=1}^{M} \sim q(y) \text{ (noise distribution)}.$$

Energy-Based Probabilistic Regression - Training using NCE

$$J_{\mathrm{NCE}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} J_{\mathrm{NCE}}^{(i)}(\theta), \quad J_{\mathrm{NCE}}^{(i)}(\theta) = \log \frac{\exp\{f_\theta(x_i, y_i^{(0)}) - \log q(y_i^{(0)})\}}{\sum_{m=0}^{M} \exp\{f_\theta(x_i, y_i^{(m)}) - \log q(y_i^{(m)})\}},$$

$$y_i^{(0)} \triangleq y_i, \quad \{y_i^{(m)}\}_{m=1}^{M} \sim q(y) \text{ (noise distribution).}$$

Effectively, $J_{\mathrm{NCE}}(\theta)$ is the softmax cross-entropy loss for a classification problem with $M+1$ classes (which of the $M+1$ values $\{y_i^{(m)}\}_{m=0}^{M}$ is the true target y_i?).

Energy-Based Probabilistic Regression - Training using NCE

$$J_{\mathrm{NCE}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} J_{\mathrm{NCE}}^{(i)}(\theta), \quad J_{\mathrm{NCE}}^{(i)}(\theta) = \log \frac{\exp\{f_\theta(x_i, y_i^{(0)}) - \log q(y_i^{(0)})\}}{\sum_{m=0}^{M} \exp\{f_\theta(x_i, y_i^{(m)}) - \log q(y_i^{(m)})\}},$$

$$y_i^{(0)} \triangleq y_i, \quad \{y_i^{(m)}\}_{m=1}^{M} \sim q(y) \text{ (noise distribution)}.$$

Effectively, $J_{\mathrm{NCE}}(\theta)$ is the softmax cross-entropy loss for a classification problem with $M+1$ classes (which of the $M+1$ values $\{y_i^{(m)}\}_{m=0}^{M}$ is the true target y_i?).

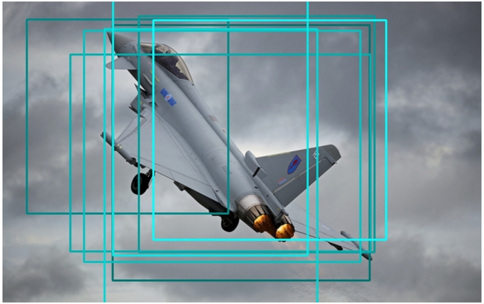
In **[Paper II] How to Train Your Energy-Based Model for Regression**, the noise distribution $q(y)$ was set to a mixture of K Gaussians centered at the true target y_i,

$$q(y) = \frac{1}{K}\sum_{k=1}^{K} \mathcal{N}(y; y_i, \sigma_k^2 I).$$

UPPSALA UNIVERSITET

$$J_{\text{NCE}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} J_{\text{NCE}}^{(i)}(\theta), \quad J_{\text{NCE}}^{(i)}(\theta) = \log \frac{\exp\{f_\theta(x_i, y_i^{(0)}) - \log q(y_i^{(0)})\}}{\sum_{m=0}^{M} \exp\{f_\theta(x_i, y_i^{(m)}) - \log q(y_i^{(m)})\}},$$

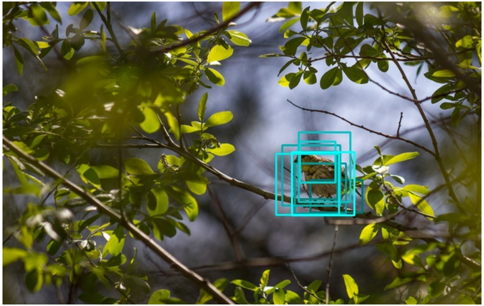
$$y_i^{(0)} \triangleq y_i, \quad \{y_i^{(m)}\}_{m=1}^{M} \sim q(y), \quad q(y) = \frac{1}{K}\sum_{k=1}^{K} \mathcal{N}(y; y_i, \sigma_k^2 I).$$



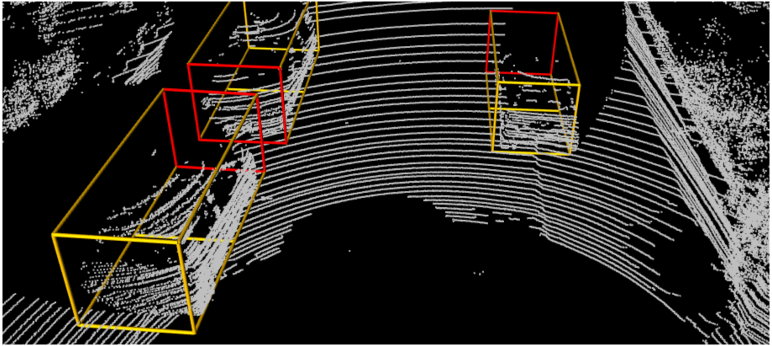
UPPSALA UNIVERSITET

$$J_{\text{NCE}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} J_{\text{NCE}}^{(i)}(\theta), \quad J_{\text{NCE}}^{(i)}(\theta) = \log \frac{\exp\{f_\theta(x_i, y_i^{(0)}) - \log q(y_i^{(0)})\}}{\sum_{m=0}^{M} \exp\{f_\theta(x_i, y_i^{(m)}) - \log q(y_i^{(m)})\}},$$

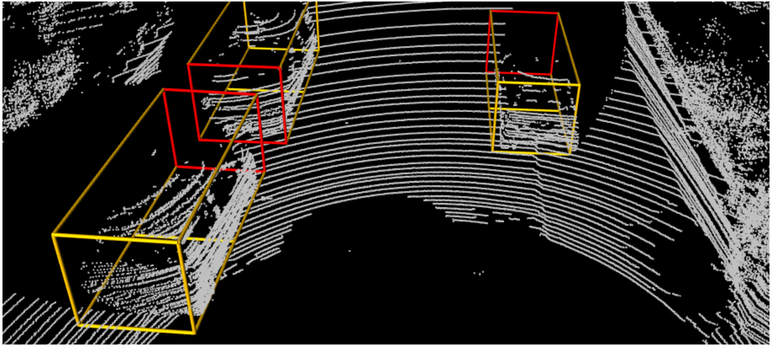
$$y_i^{(0)} \triangleq y_i, \quad \{y_i^{(m)}\}_{m=1}^{M} \sim q(y), \quad q(y) = \frac{1}{K}\sum_{k=1}^{K} \mathcal{N}(y; y_i, \sigma_k^2 I).$$



Energy-Based Probabilistic Regression - 3D Object Detection

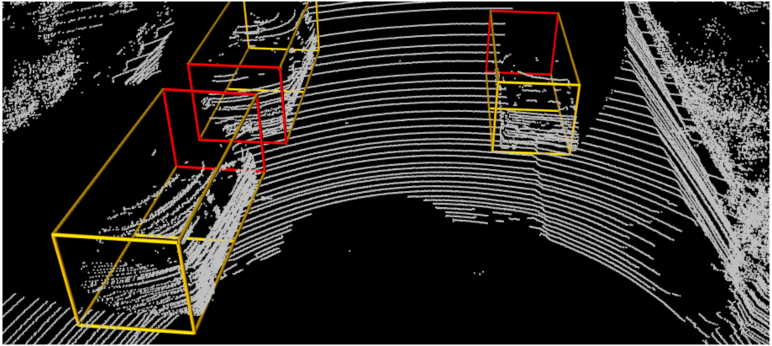


Energy-Based Probabilistic Regression - 3D Object Detection



In **[Paper III] Accurate 3D Object Detection using Energy-Based Models**, we extend our energy-based regression approach from 2D to 3D bounding box regression.

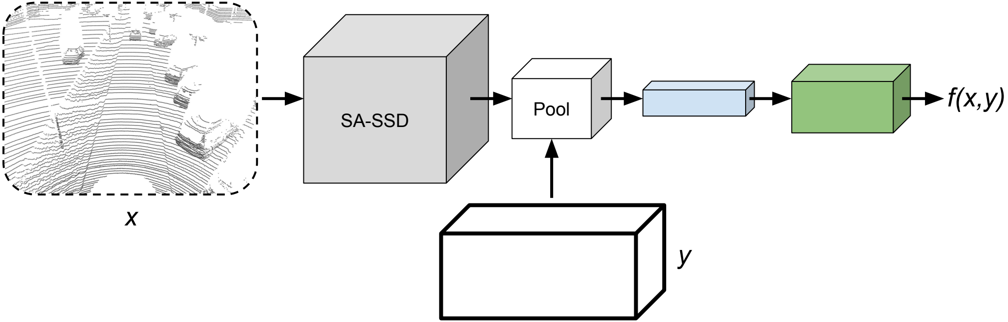
Energy-Based Probabilistic Regression - 3D Object Detection

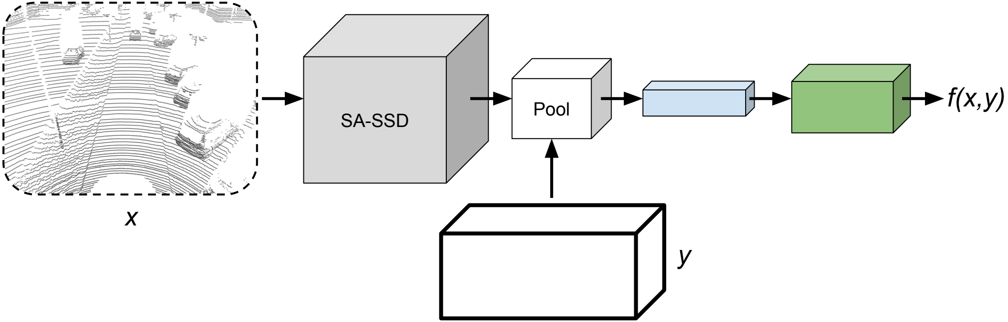


In **[Paper III] Accurate 3D Object Detection using Energy-Based Models**, we extend our energy-based regression approach from 2D to 3D bounding box regression.

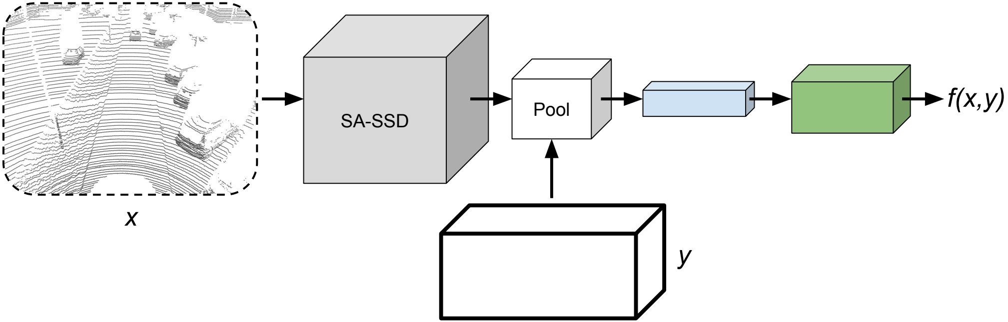
This is achieved by designing a differentiable pooling operator for 3D bounding boxes $y \in \mathbb{R}^7$, and adding an extra network branch to a state-of-the-art 3D object detector.

Energy-Based Probabilistic Regression - 3D Object Detection





We integrate a conditional EBM $p(y|x;\theta) = e^{f_\theta(x,y)} / \int e^{f_\theta(x,\tilde{y})} d\tilde{y}$ into the SA-SSD 3D object detector.



We integrate a conditional EBM $p(y|x;\theta) = e^{f_\theta(x,y)} / \int e^{f_\theta(x,\tilde{y})} d\tilde{y}$ into the SA-SSD 3D object detector.

We design a differentiable pooling operator that, given a 3D bounding box y, extracts a feature vector from the SA-SSD output. This feature vector is processed by fully-connected layers, outputting $f_\theta(x, y) \in \mathbb{R}$.

UPPSALA UNIVERSITET

In **Paper I**, we trained the EBM $p(y|x;\theta)$ by approximating the negative log-likelihood $\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log p(y_i|x_i;\theta)$ using importance sampling:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{1}{M}\sum_{m=1}^{M} \frac{e^{f_\theta(x_i, y_i^{(m)})}}{q(y_i^{(m)})}\right) - f_\theta(x_i, y_i),$$

$$\{y_i^{(m)}\}_{m=1}^{M} \sim q(y) \text{ (proposal distribution).}$$

Energy-Based Probabilistic Regression - Practical Limitations

In **Paper II & III**, we trained the EBM $p(y|x;\theta)$ using NCE:

$$J_{\mathrm{NCE}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} J_{\mathrm{NCE}}^{(i)}(\theta), \quad J_{\mathrm{NCE}}^{(i)}(\theta) = \log \frac{\exp\{f_\theta(x_i, y_i^{(0)}) - \log q(y_i^{(0)})\}}{\sum_{m=0}^{M} \exp\{f_\theta(x_i, y_i^{(m)}) - \log q(y_i^{(m)})\}},$$

$$y_i^{(0)} \triangleq y_i, \quad \{y_i^{(m)}\}_{m=1}^{M} \sim q(y) \text{ (noise distribution).}$$

Energy-Based Probabilistic Regression - Practical Limitations

In **Paper II & III**, we trained the EBM $p(y|x;\theta)$ using NCE:

$$J_{\mathrm{NCE}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} J_{\mathrm{NCE}}^{(i)}(\theta), \quad J_{\mathrm{NCE}}^{(i)}(\theta) = \log \frac{\exp\{f_\theta(x_i, y_i^{(0)}) - \log q(y_i^{(0)})\}}{\sum_{m=0}^{M} \exp\{f_\theta(x_i, y_i^{(m)}) - \log q(y_i^{(m)})\}},$$

$$y_i^{(0)} \triangleq y_i, \quad \{y_i^{(m)}\}_{m=1}^{M} \sim q(y) \text{ (noise distribution).}$$

In both cases, the proposal/noise distribution $q(y)$ was set to a mixture of K Gaussian components centered at the true target y_i,

$$q(y) = \frac{1}{K}\sum_{k=1}^{K} \mathcal{N}(y; y_i, \sigma_k^2 I).$$

UPPSALA UNIVERSITET

$$\{y_i^{(m)}\}_{m=1}^{M} \sim q(y), \quad q(y) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(y; y_i, \sigma_k^2 I)$$

Energy-Based Probabilistic Regression - Practical Limitations

UPPSALA UNIVERSITET

$$\{y_i^{(m)}\}_{m=1}^{M} \sim q(y), \quad q(y) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(y; y_i, \sigma_k^2 I)$$

The proposal/noise distribution $q(y)$ contains hyperparameters K and $\{\sigma_k^2\}_{k=1}^{K}$, which need to be carefully tuned for each specific problem.

$$\{y_i^{(m)}\}_{m=1}^{M} \sim q(y), \quad q(y) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(y; y_i, \sigma_k^2 I)$$

The proposal/noise distribution $q(y)$ contains hyperparameters K and $\{\sigma_k^2\}_{k=1}^{K}$, which need to be carefully tuned for each specific problem.

The proposal/noise distribution $q(y)$ depends on the true target y_i, and can therefore only be utilized during training.

$$\{y_i^{(m)}\}_{m=1}^{M} \sim q(y), \quad q(y) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(y; y_i, \sigma_k^2 I)$$

The proposal/noise distribution $q(y)$ contains hyperparameters K and $\{\sigma_k^2\}_{k=1}^{K}$, which need to be carefully tuned for each specific problem.

The proposal/noise distribution $q(y)$ depends on the true target y_i, and can therefore only be utilized during training.

- To produce a prediction $y^\star$ at test-time, **Paper I, II & III** employed gradient ascent to refine an initial estimate $\hat{y}$. However, this prediction strategy then requires access to good initial estimates.

Energy-Based Probabilistic Regression - Practical Limitations

$q(y)$ contains task-dependent hyperparameters K and $\{\sigma_k^2\}_{k=1}^{K}$.

$q(y)$ depends on the true target y_i and can thus only be utilized during training.

$q(y)$ contains task-dependent hyperparameters K and $\{\sigma_k^2\}_{k=1}^K$.

$q(y)$ depends on the true target y_i and can thus only be utilized during training.

In **[Paper IV] Learning Proposals for Practical Energy-Based Regression**, we address both these limitations by jointly learning a parameterized proposal/noise distribution $q(y|x;\phi)$ during EBM training.

$q(y)$ contains task-dependent hyperparameters K and $\{\sigma_k^2\}_{k=1}^K$.

$q(y)$ depends on the true target y_i and can thus only be utilized during training.

In **[Paper IV] Learning Proposals for Practical Energy-Based Regression**, we address both these limitations by jointly learning a parameterized proposal/noise distribution $q(y|x;\phi)$ during EBM training.

We derive an efficient and convenient objective that can be employed to train $q(y|x;\phi)$ by directly minimizing its KL divergence to the EBM $p(y|x;\theta)$.

We want $q(y|x;\phi)$ to be a close approximation of the EBM $p(y|x;\theta)$. Specifically, we want to find ϕ that minimizes the KL divergence between $q(y|x;\phi)$ and $p(y|x;\theta)$.

We want $q(y|x;\phi)$ to be a close approximation of the EBM $p(y|x;\theta)$. Specifically, we want to find ϕ that minimizes the KL divergence between $q(y|x;\phi)$ and $p(y|x;\theta)$.

Therefore, we seek to compute $\nabla_\phi D_{\mathrm{KL}}\big(p(y|x;\theta) \parallel q(y|x;\phi)\big)$. The gradient $\nabla_\phi D_{\mathrm{KL}}$ is generally intractable, but can be conveniently approximated by the following result:

Practical Energy-Based Regression - Learning the Proposal

We want $q(y|x;\phi)$ to be a close approximation of the EBM $p(y|x;\theta)$. Specifically, we want to find ϕ that minimizes the KL divergence between $q(y|x;\phi)$ and $p(y|x;\theta)$.

Therefore, we seek to compute $\nabla_\phi D_{\mathrm{KL}}\big(p(y|x;\theta) \parallel q(y|x;\phi)\big)$. The gradient $\nabla_\phi D_{\mathrm{KL}}$ is generally intractable, but can be conveniently approximated by the following result:

Result 1: For a conditional EBM $p(y|x;\theta) = e^{f_\theta(x,y)} / \int e^{f_\theta(x,\tilde{y})} d\tilde{y}$ and distribution $q(y|x;\phi)$,

$$\nabla_\phi D_{\mathrm{KL}}\big(p \parallel q\big) \approx \nabla_\phi \log\left(\frac{1}{M}\sum_{m=1}^{M} \frac{e^{f_\theta(x,y^{(m)})}}{q(y^{(m)}|x;\phi)}\right),$$

where $\{y^{(m)}\}_{m=1}^{M}$ are M independent samples drawn from $q(y|x;\phi)$.

Result 1: For a conditional EBM $p(y|x;\theta) = e^{f_\theta(x,y)} / \int e^{f_\theta(x,\tilde{y})} d\tilde{y}$ and distribution $q(y|x;\phi)$,

$$\nabla_\phi D_{\mathrm{KL}}\big(p \parallel q\big) \approx \nabla_\phi \log \left(\frac{1}{M} \sum_{m=1}^{M} \frac{e^{f_\theta(x,y^{(m)})}}{q(y^{(m)}|x;\phi)} \right),$$

where $\{y^{(m)}\}_{m=1}^{M}$ are M independent samples drawn from $q(y|x;\phi)$.

Practical Energy-Based Regression - Learning the Proposal

Result 1: For a conditional EBM $p(y|x;\theta) = e^{f_\theta(x,y)} / \int e^{f_\theta(x,\tilde{y})} d\tilde{y}$ and distribution $q(y|x;\phi)$,

$$\nabla_\phi D_{\mathrm{KL}}\big(p \parallel q\big) \approx \nabla_\phi \log\left(\frac{1}{M}\sum_{m=1}^{M} \frac{e^{f_\theta(x,y^{(m)})}}{q(y^{(m)}|x;\phi)}\right),$$

where $\{y^{(m)}\}_{m=1}^{M}$ are M independent samples drawn from $q(y|x;\phi)$.

Given data $\{x_i\}_{i=1}^{N}$, Result 1 implies that the proposal/noise distribution $q(y|x;\phi)$ can be trained to approximate the EBM $p(y|x;\theta)$ by minimizing the loss,

$$J_{\mathrm{KL}}(\phi) = \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{1}{M}\sum_{m=1}^{M} \frac{e^{f_\theta(x_i,y_i^{(m)})}}{q(y_i^{(m)}|x_i;\phi)}\right),$$

where $\{y_i^{(m)}\}_{m=1}^{M} \sim q(y|x_i;\phi)$.

Practical Energy-Based Regression - Joint Training Method

Given data $\{x_i\}_{i=1}^N$, Result 1 implies that the proposal/noise distribution $q(y|x;\phi)$ can be trained to approximate the EBM $p(y|x;\theta)$ by minimizing the loss,

$$J_{\mathrm{KL}}(\phi) = \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{1}{M}\sum_{m=1}^{M} \frac{e^{f_\theta(x_i, y_i^{(m)})}}{q(y_i^{(m)}|x_i;\phi)}\right),$$

where $\{y_i^{(m)}\}_{m=1}^M \sim q(y|x_i;\phi)$.

Given data $\{x_i\}_{i=1}^N$, Result 1 implies that the proposal/noise distribution $q(y|x;\phi)$ can be trained to approximate the EBM $p(y|x;\theta)$ by minimizing the loss,

$$J_{\mathrm{KL}}(\phi) = \frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{1}{M}\sum_{m=1}^{M}\frac{e^{f_\theta(x_i, y_i^{(m)})}}{q(y_i^{(m)}|x_i;\phi)}\right),$$

where $\{y_i^{(m)}\}_{m=1}^M \sim q(y|x_i;\phi)$.

Note that $J_{\mathrm{KL}}(\phi)$ is identical to the first term of the EBM loss $J(\theta)$ from **Paper I**,

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{1}{M}\sum_{m=1}^{M}\frac{e^{f_\theta(x_i, y_i^{(m)})}}{q(y_i^{(m)})}\right) - f_\theta(x_i, y_i).$$

UPPSALA UNIVERSITET

Given data $\{x_i\}_{i=1}^N$, Result 1 implies that the proposal/noise distribution $q(y|x;\phi)$ can be trained to approximate the EBM $p(y|x;\theta)$ by minimizing the loss,

$$J_{\mathrm{KL}}(\phi) = \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{1}{M}\sum_{m=1}^{M} \frac{e^{f_\theta(x_i, y_i^{(m)})}}{q(y_i^{(m)}|x_i;\phi)}\right),$$

where $\{y_i^{(m)}\}_{m=1}^M \sim q(y|x_i;\phi)$.

Note that $J_{\mathrm{KL}}(\phi)$ is identical to the first term of the EBM loss $J(\theta)$ from **Paper I**,

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{1}{M}\sum_{m=1}^{M} \frac{e^{f_\theta(x_i, y_i^{(m)})}}{q(y_i^{(m)})}\right) - f_\theta(x_i, y_i).$$

$J(\theta)$ can thus be used as a joint objective for training both q and the EBM p.

Practical Energy-Based Regression - Joint Training Method

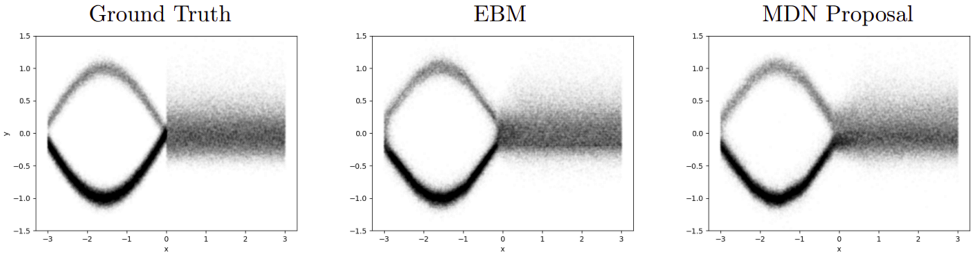
$$J(\theta, \phi) = \frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{1}{M} \sum_{m=1}^{M} \frac{e^{f_\theta(x_i, y_i^{(m)})}}{q(y_i^{(m)}|x_i; \phi)} \right) - f_\theta(x_i, y_i),$$

$$\{y_i^{(m)}\}_{m=1}^{M} \sim q(y|x_i; \phi).$$

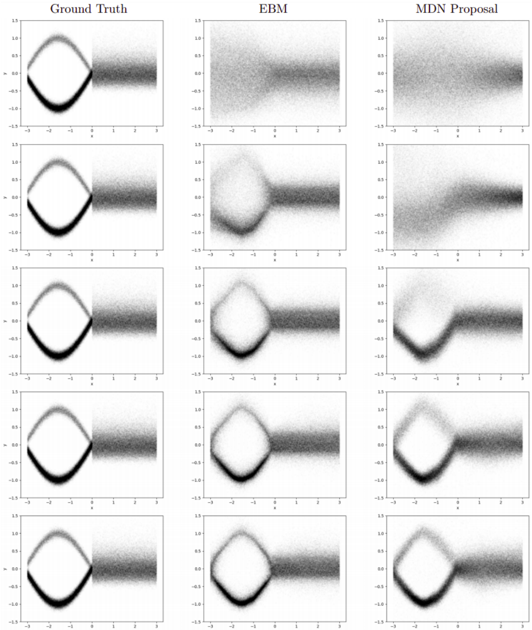
$$J(\theta, \phi) = \frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{1}{M} \sum_{m=1}^{M} \frac{e^{f_\theta(x_i, y_i^{(m)})}}{q(y_i^{(m)}|x_i; \phi)} \right) - f_\theta(x_i, y_i),$$

$$\{y_i^{(m)}\}_{m=1}^{M} \sim q(y|x_i; \phi).$$

The EBM $p(y|x;\theta)$ and proposal $q(y|x;\phi)$ can be trained by jointly minimizing $J(\theta,\phi)$ w.r.t. both θ and ϕ:



Practical Energy-Based Regression - Joint Training Method



$$J_{\mathrm{KL}}(\phi) = \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{1}{M}\sum_{m=1}^{M} \frac{e^{f_\theta(x_i, y_i^{(m)})}}{q(y_i^{(m)}|x_i;\phi)}\right),$$

$$J_{\mathrm{NCE}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} J_{\mathrm{NCE}}^{(i)}(\theta), \quad J_{\mathrm{NCE}}^{(i)}(\theta) = \log \frac{\exp\{f_\theta(x_i, y_i^{(0)}) - \log q(y_i^{(0)})\}}{\sum_{m=0}^{M} \exp\{f_\theta(x_i, y_i^{(m)}) - \log q(y_i^{(m)})\}},$$

$$y_i^{(0)} \triangleq y_i, \quad \{y_i^{(m)}\}_{m=1}^{M} \sim q(y|x_i;\phi).$$

Practical Energy-Based Regression - Joint Training Method

$$J_{\mathrm{KL}}(\phi) = \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{1}{M}\sum_{m=1}^{M} \frac{e^{f_\theta(x_i, y_i^{(m)})}}{q(y_i^{(m)}|x_i;\phi)}\right),$$

$$J_{\mathrm{NCE}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} J_{\mathrm{NCE}}^{(i)}(\theta), \quad J_{\mathrm{NCE}}^{(i)}(\theta) = \log \frac{\exp\{f_\theta(x_i, y_i^{(0)}) - \log q(y_i^{(0)})\}}{\sum_{m=0}^{M} \exp\{f_\theta(x_i, y_i^{(m)}) - \log q(y_i^{(m)})\}},$$

$$y_i^{(0)} \triangleq y_i, \quad \{y_i^{(m)}\}_{m=1}^{M} \sim q(y|x_i;\phi).$$

The EBM $p(y|x;\theta)$ and proposal/noise distribution $q(y|x;\phi)$ can also be jointly trained by updating ϕ via the loss $J_{\mathrm{KL}}(\phi)$, and updating θ via $J_{\mathrm{NCE}}(\theta)$.

Practical Energy-Based Regression - Utilizing the Proposal

UPPSALA UNIVERSITET

We have access to a proposal $q(y|x;\phi)$ that is conditioned only on the input x and thus can be utilized also at test-time.

We have access to a proposal $q(y|x;\phi)$ that is conditioned only on the input x and thus can be utilized also at test-time.

As $q(y|x;\phi)$ has been trained to approximate the EBM $p(y|x;\theta)$, it can be utilized with self-normalized importance sampling to approximate expectations $\mathbb{E}_p$ w.r.t. the EBM,

$$\mathbb{E}_p[\xi(y)] = \int \xi(y) p(y|x;\theta) dy \approx \sum_{m=1}^{M} w^{(m)} \xi\big(y^{(m)}\big),$$

$$w^{(m)} = \frac{e^{f_\theta(x, y^{(m)})} / q(y^{(m)}|x;\phi)}{\sum_{l=1}^{M} e^{f_\theta(x, y^{(l)})} / q(y^{(l)}|x;\phi)},$$

where $\{y^{(m)}\}_{m=1}^{M} \sim q(y|x;\phi)$.

Practical Energy-Based Regression - Utilizing the Proposal

We have access to a proposal $q(y|x;\phi)$ that is conditioned only on the input x and thus can be utilized also at test-time.

As $q(y|x;\phi)$ has been trained to approximate the EBM $p(y|x;\theta)$, it can be utilized with self-normalized importance sampling to approximate expectations $\mathbb{E}_p$ w.r.t. the EBM,

$$\mathbb{E}_p[\xi(y)] = \int \xi(y) p(y|x;\theta) dy \approx \sum_{m=1}^{M} w^{(m)} \xi\big(y^{(m)}\big),$$

$$w^{(m)} = \frac{e^{f_\theta(x, y^{(m)})} / q(y^{(m)}|x;\phi)}{\sum_{l=1}^{M} e^{f_\theta(x, y^{(l)})} / q(y^{(l)}|x;\phi)},$$

where $\{y^{(m)}\}_{m=1}^{M} \sim q(y|x;\phi)$.

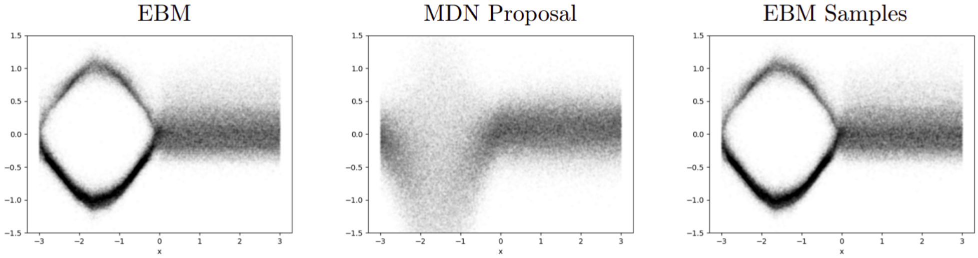
Setting $\xi(y) = y$ enables us to approximately compute the EBM mean. In this manner, we can thus directly produce a stand-alone prediction $y^\star$ for the EBM $p(y|x;\theta)$.

Practical Energy-Based Regression - Utilizing the Proposal

$$w^{(m)} = \frac{e^{f_\theta(x, y^{(m)})}/q(y^{(m)}|x;\phi)}{\sum_{l=1}^{M} e^{f_\theta(x, y^{(l)})}/q(y^{(l)}|x;\phi)}$$

$$w^{(m)} = \frac{e^{f_\theta(x, y^{(m)})} / q(y^{(m)}|x; \phi)}{\sum_{l=1}^{M} e^{f_\theta(x, y^{(l)})} / q(y^{(l)}|x; \phi)}$$

We can also draw approximate samples from the EBM $p(y|x; \theta)$ by re-sampling with replacement from the set $\{y^{(m)}\}_{m=1}^{M} \sim q(y|x; \phi)$ of proposal samples, drawing each $y^{(m)}$ with probability $w^{(m)}$:



Summary

Energy-based models.

Energy-based models for regression.

How to train energy-based models for regression.

- Noise contrastive estimation (NCE).

Energy-based regression for 3D object detection.

Practical limitations of energy-based regression.

Learning proposals for more practical energy-based regression.

Contact

UPPSALA UNIVERSITET

Fredrik K. Gustafsson, Uppsala University

fredrik.gustafsson@it.uu.se

www.fregu856.com